Quantum-ESPRESSO

opEn-Source Package for Research in Electronic Structure, Simulation, and Optimization

Quantum ESPRESSO

http://www.quantum-espresso.org/

QUANTUMESPRESSO

HOME ::PROJECT ::WHAT CAN QE DO ::DOWNLOAD ::LEARN ::PSEUDO ::TOOLS ::
QE WIKI ::CONTACTS ::QUOTE ::LOGOS ::

28 July 2011 Version 4.3.2 of Quantum ESPRESSO is available for download.

25 May 2011
Version 4.3.1 of Quantum ESPRESSO is available for download.

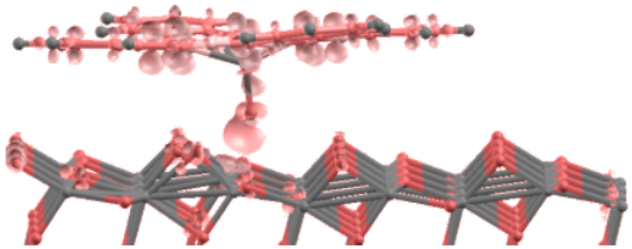
05 May 2011
The first GPU-enabled beta release of Quantum ESPRESSO is available for download.

01 April 2011
The new release, v.4.3, of the Quantum ESPRESSO distribution is available for download.

13 July 2010
Bugfix release v.4.2.1 of the Quantum ESPRESSO distribution is available for download.

10 May 2010
A new version, v.4.2, of the Quantum ESPRESSO distribution is available for download.

Quantum ESPRESSO is an integrated suite of computer codes for electronic-structure calculations and materials modeling at the nanoscale. It is based on density-functional theory, plane waves, and pseudopotentials (both norm-conserving and ultrasoft).



What I cannot compute, I do not understand [adapted from Richard P. Feynman]

Other Useful Packages

FFTW

http://www.fftw.org/

LAPACK — Linear Algebra PACKage

http://www.netlib.org/lapack/

BLAS (Basic Linear Algebra Subprograms)

http://www.netlib.org/blas/

http://www.s3.infm.it/iotk/

Quantum-ESPRESSO

- **Quantum-ESPRESSO**
  - An open source full ab-initio package implementation of electronic structure and energy calculations, linear response methods (to calculate phonon dispersion curves, dielectric constants, and Born effective charges) and third-order anharmonic perturbation theory.
  - Contains two molecular-dynamics codes, CPMD (Car-Parrinello Molecular Dynamics) and FPMD (First-Principles Molecular Dynamics).
  - PWSCF (Plane-Wave Self-Consistent Field)
    - We will use this to perform total energy calculations.
    - It uses both norm-conserving pseudopotentials (PP) and ultrasoft pseudopotentials (US-PP), within the density functional theory (DFT).
- This is a tutorial on how to get energies using PWSCF code in Quantum-Espresso.

Useful References
http://www.quantum-espresso.org/user_guide.pdf
http://www.quantum-espresso.org/user_guide/user_guide.html
Doc folder of the package contain html pages having neatly created documentation

Quantum-ESPRESSO

Installation

1.Download the latest sources from the website. ESPRESSO comes with an inbuilt library for LAPACK and BLAS.

2.It is recommended that you download FFTW as well and if you have a multi-core system use mpi compilers to take advantage of it.

3.The installation from source follows the standard procedure of

1. configure, make

4.Use the options of configure to help it libraries

1. ./configure –help

5.There is a gui version also.

Spending time on a proper installation will save a lot of time during run time!!!

Atomic Units

- The package uses Atomic Units
- Some helpful conversions are provided:

1 **bohr** = 1 **a.u.** (atom unit) = 0.529177249 **angstroms**
1 **Rydberg** (Ry) = 13.6056981 **eV**
1 **eV** = 1.60217733 x 10^{-19} **Joules**

- You may also consult the following on-line conversion link:
  - http://www.translatorscafe.com/cafe/units-converter/magnetic-flux/c/

PWscf

- One can perform Plane Wave self consistent field calculations for DFT. The executable is called pw.x
- Features
  - regimented input
  - primitive cell representation
  - pseudopotentials
- Refer Documentation
  - INPUT_PW.html

Pseudopotentials
You can find them on
http://www.quantum-espresso.org/pseudo.php

Input File Description

Program: pw.x / PWscf / Quantum Espresso

TABLE OF CONTENTS

INPUT file in PWscf

The input file for PWscf is structured in a number of NAMELISTS and INPUT_CARDS.

```
 &NAMELIST1 ... /

&NAMELIST2 ... /

&NAMELIST3 ... /

INPUT_CARD1
 ....
 ....
INPUT_CARD2
 ....
 ....
```

There are **three mandatory** NAMELISTS in PWscf:

&CONTROL: control the flux of the calculation and the amount of I/O on disk and on the screen.

&SYSTEM: specify the system under study.

&ELECTRONS: control the algorithms used to reach the self-consistent solution of KS equations for the electrons.

and **three mandatory** INPUT_CARDS in PWscf:

ATOMIC_SPECIES: name, mass and pseudopotential used for each atomic species.

ATOMIC_POSITION: type and coordinates of each atom.

K_POINTS: coordinates and weights of the k-points used for Brillouin Zone integration.

An example input

```
(1)     &control
(2)         calculation = 'scf'
(3)         restart_mode='from_scratch'
(4)         prefix='diamond'
(5)         tstress = .true.
(6)         tprnfor = .true.
(7)         pseudo_dir = '/state/partition1/marzari'
(8)         outdir = '/state/partition1/marzari'
(9)     /
```

Lines numbers are added for reference.

Line 1: `&control` declares the control block.

Line 2: `calculation = 'scf'` tells PWSCF that this will be a self-consistent field calculation.

Line 3: `restart_mode = 'from_scratch'`, declares that we will be generating a new structure.

Line 4:

`prefix='diamond'`, declares the filename prefix to be used for temporary files.

An example input

```
(1)     &control
(2)         calculation = 'scf'
(3)         restart_mode='from_scratch'
(4)         prefix='diamond'
(5)         tstress = .true.
(6)         tprnfor = .true.
(7)         pseudo_dir = '/state/partition1/marzari'
(8)         outdir = '/state/partition1/marzari'
(9)     /
```

Line 5:

`tstress = .true.` is a flag to calculate the stresses.

Line 6:

`tprnfor = .true.` is a flag to calculate the forces.

Line 7:

`pseudo_dir = '/state/partition1/marzari'`, defines the location of the directory where you store the pseudo-potentials.

Line 8:

`outdir='/state/partition1/marzari'` defines the location of the temporary files. This should always be a local scratch disk so that large I/O operations do not occur across the network.

An example input

```
(10)    &system
(11)        ibrav=  2, celldm(1) =6.60, nat=  2, ntyp= 1
(12)        ecutwfc =40
(13)    /
```

Note: When using ultrasoft pseudo potentials use the keyword ecutrho.

ecutrho should be set to 8 or 12 times ecutwfc

Lines 10-13: the system block ibrav – gives the crystal system. ibrav=2 is a face-centered cubic structure. This is used because the symmetry of the structure can reduce the number of calculations you need to do. If you need other crystal systems, consult the INPUT_PW file .

celldm – defines the dimensions of the cell. You will be changing this parameter. celldm is in atomic units, or bohrs. Remember that 1 bohr = 0.529177249 angstroms. The value will depend on the Bravais lattice of the structure (see below for a key). For FCC, celldm(1) = $^{a}0$. In cubic systems, a=b=c= $^{a}0$. Consult INPUT_PW for other crystal systems.

nat – number of atoms (each individual unique atom). Note that diamond has two unique atoms in the smallest assymetric unit.

ntyp – number of types of atoms

ecutwfc – Energy cutoff for pseudo-potentials. This one is important; you will be changing this parameter.

An example input

```
(14)    &electrons
(15)        diagonalization='david'
(16)        mixing_mode = 'plain'
(17)        mixing_beta = 0.7
(18)        conv_thr =  1.0d-8
(19)        /
```

Lines 14-19: The electrons block

`diagonalization` – diagonalization method. Use the default for now.

`mixing_beta` - Mixing factor. Don't worry about this for now.

`mixing_mode` – Mixing method. Don't worry about this for now.

`conv_thr` – Convergence threshold. Use the default for now.

An example input

```
(20)    ATOMIC_SPECIES
(21)        C  12.011  C.pz-vbc.UPF
(22)    ATOMIC_POSITIONS
(23)        C 0.00 0.00 0.00
(24)        C 0.25 0.25 0.25
(25)    K_POINTS {automatic}
(26)        4 4 4 0 0 0
```

The syntax to run the files in non-interactive mode is (using I/O redirection)

pw.x <input_file >output_file

Lines 20-21: Atomic species declaration
After the keyword ATOMIC_SPECIES, for each ntyp enter
atomic symbol atomic weight pseudo-potential

Lines 22-24: Atomic positions
After the keyword ATOMIC_POSITIONS, for each nat enter
atomic symbol x y z
where x,y,z are given as fractional coordinates of the conventional cell.

Lines 25-26: k-point selection
after the keywork K_POINTS, "automatic" tells PWSCF to automatically generate a k-point grid.
The format of the next line is
nkx nky nkz offx offy offz
where nk* is the number of intervals in a direction and off* is the offset of the origin of the grid.

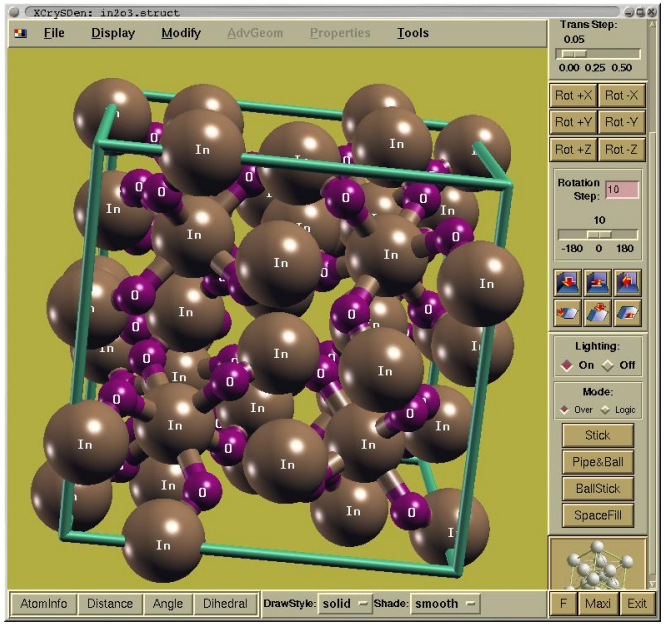
Setting up the unit cell

- Setting up the structure is generally the hardest
- It is recommended that you use a package such as XCrySDen to visualize the structure once you have plugged it in the PWscf input file.
  - Download it from http://www.xcrysden.org/
  - It can be used to open PWscf input files
- The proper choice of Bravais lattice will ensure utilization of symmetry. Utilizing symmetry reduces computation time significantly.

Two Essential Steps

Forming the Bravais Lattice

Positioning the Atoms



Bravais Lattice

- The “ibrav” parameter helps in selecting this. There are 15 possible choices

```
ibrav Bravais-lattice index (NO default, must be specified)


  0      read unit cell information from CELL_PARAMETRES card
  1      cubic P (sc)
  2      cubic F (fcc)
  3      cubic I (bcc)
  4      Hexagonal and Trigonal P
  5      Trigonal R
  6      Tetragonal P (st)
  7      Tetragonal I (bct)
  8      Orthorhombic P
  9      Orthorhombic base-centered(bco)
 10      Orthorhombic face-centered
 11      Orthorhombic body-centered
 12      Monoclinic P
 13      Monoclinic base-centered
 14      Triclinic P
```

Bravais Lattice

- Then you need to specify the crystallographic constants: there are two options:

```
1)  celldm(i), i=1,2,...,6
2)  a, b, c, cosab, cosac, cosbc,
```

```
celldm(1) = a / bohr_radius_angs = alat (internal unit of lenght)
celldm(2) = b / a
celldm(3) = c / a
celldm(4) = cosab
celldm(5) = cosac
celldm(6) = cosbc
```

```
BEWARE:
       alat = celldm(1) is the lattice parameter "a" in BOHR
       a,b,c are given in ANGSTROM
```

- Specify either a, b, c, **OR** celldm **but not both**. Crystallographic constants are to be specified for the Bravais lattice used; other parameters are **ignored**.

Bravais Lattice

ibrav	celldm -->	a 1	b 2	c 3	cosab 4	cosac 5	cosbc 6
1	cubic P (sc)	*					
2	cubic F (fcc)	*					
3	cubic I (bcc)	*					
4	Hexagonal and Trigonal P	*		*			
5	Trigonal R	*			*		
6	Tetragonal P (st)	*		*			
7	Tetragonal I (bct)	*		*			
8	Orthorhombic P	*	*	*			
9	Orthorhombic base-centered(bco)	*	*	*			
10	Orthorhombic face-centered	*	*	*			
11	Orthorhombic body-centered	*	*	*			
12	Monoclinic P	*	*	*	*		
13	Monoclinic base-centered	*	*	*	*		
14	Triclinic P	*	*	*	*	*	*

Bravais Lattice

- If ibrav = 0
  - Bravais Lattice fundamental vectors are read from an optional card
  - CELL_PARAMETERS to be inserted after all needed NAMELISTS.

```
&LAST_REQUIRED_NAMELIST
   ...
/
CELL_PARAMETERS  symmetry_class
   a(1,1) a(2,1) a(3,1)
   a(1,2) a(2,2) a(3,2)
   a(1,3) a(2,3) a(3,3)
```

- where *symmetry_class* is *cubic* or *hexagonal* depending on the expected symmetry of the system w.r.t the assumed reference system. It is just an indicator to help it find symmetries.
- if celldm(1) != 0, lattice vectors are in the specified units of celldm(1)
- if celldm(1) = 0, lattice vectors are given in *bohrs*.

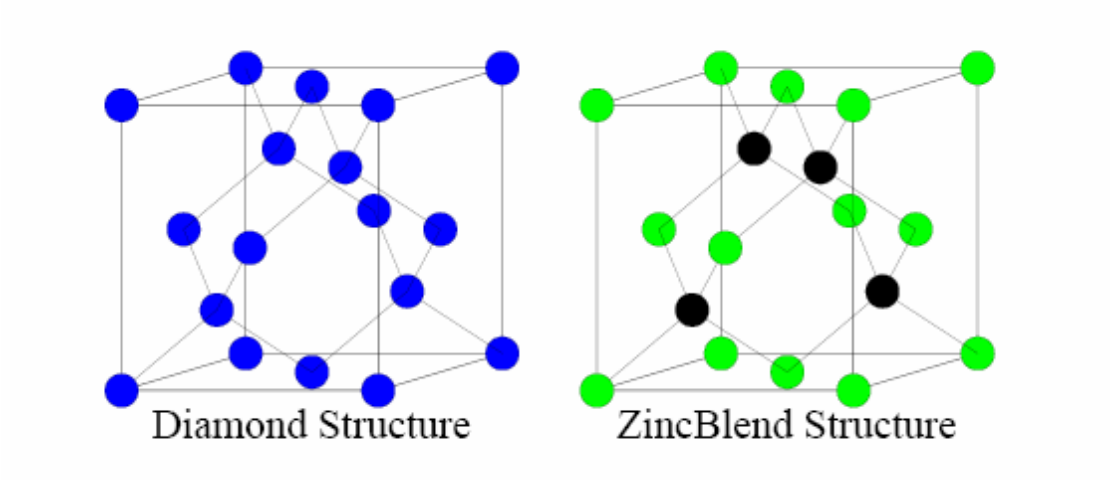
Positioning the atoms

- Atomic positions in the unit cell can be defined in 4 ways:

1. In units of alat (default) **{alat}**
2. In crystal coordinates **{crystal}**
3. In bohr {**bohr}**
4. in angstrom {**angstrom}**

Example 1: Diamond (FCC)

- How does the crystal structure look like?



- There are many ways to define the diamond structure. Here we show two.

1) Simple cubic bravais + 8 atom basis at position:

   (0 0 0),(0 ½ ½), (½ 0 ½); (½ ½ 0), (¼ ¼ ¼), (¼ ¾ ¾), (¾ ¼ ¾), (¾ ¾ ¼)

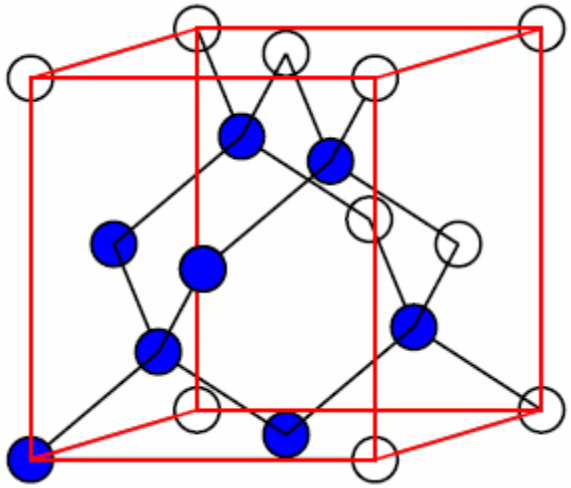
2) FCC lattice + 2 atom basis at position:

   (0 0 0), (½ ½ ½)

Example 1: Diamond (FCC)

C: simple cubic : ibrav = 1, the lattice parameter is 3.56679 A.

```
 &SYSTEM
    ntyp=1, nat=8, ibrav=1, a =  3.56679,
/
...
ATOMIC_SPECIES
C 28.086 C.UPF
ATOMIC_POSITIONS
C 0.0 0.0 0.0
C 0.0 0.5 0.5
C 0.5 0.0 0.5
C 0.5 0.5 0.0
C 0.25 0.25 0.25
C 0.25 0.75 0.75
C 0.75 0.25 0.75
C 0.75 0.75 0.25
```



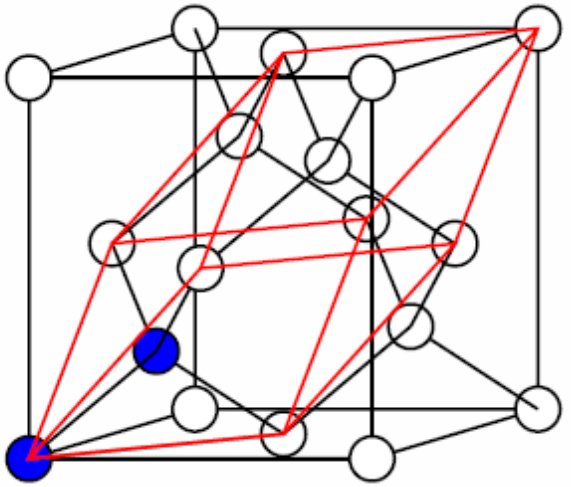
If using celldm, 3.56679 A = 6.740259 bohr, then

```
&SYSTEM
   ntyp=1, nat=8, ibrav=1, celldm(1)=6.740259,
/
```

Example 1: Diamond (FCC)

C: face centered cubic : ibrav = 2, the lattice parameter is 6.740259 bohr.

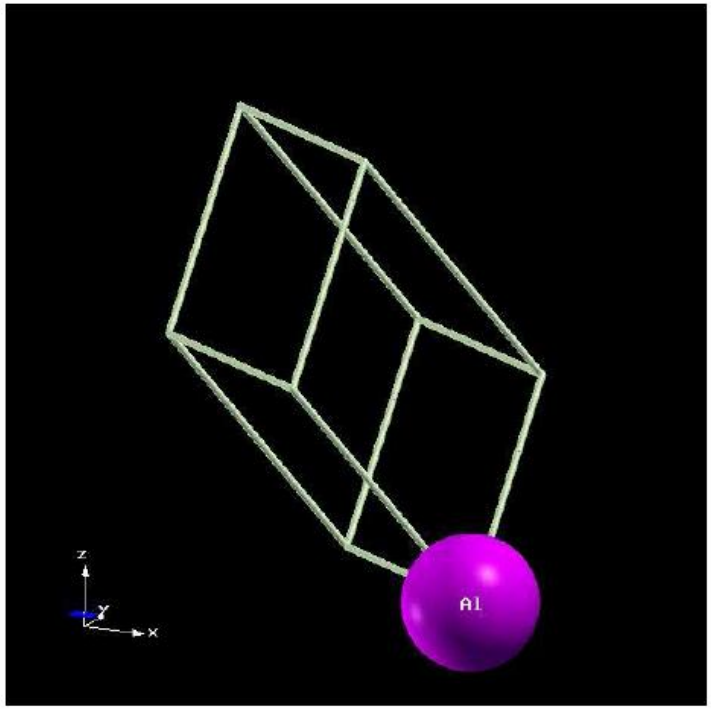
```
 &SYSTEM
    ntyp=1, nat=2, ibrav=2, celldm(1)=6.740259,
 /
...
ATOMIC_SPECIES
C 28.086 C.UPF
ATOMIC_POSITIONS
C 0.0 0.0 0.0
C 0.25 0.25 0.25
```

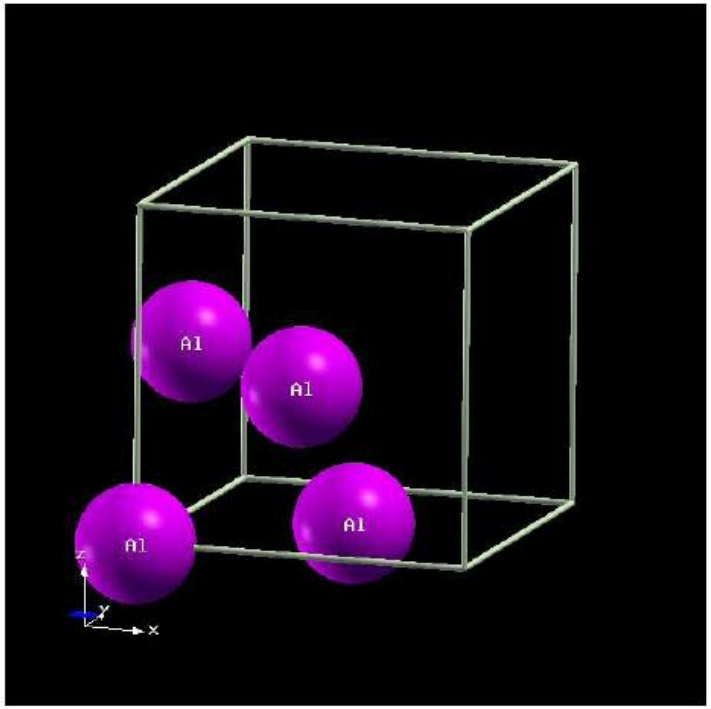


Example 2: Al (FCC)

- In this example, we will show different unit cell choice for a crystal structure and the difference between the atomic position coordinate.

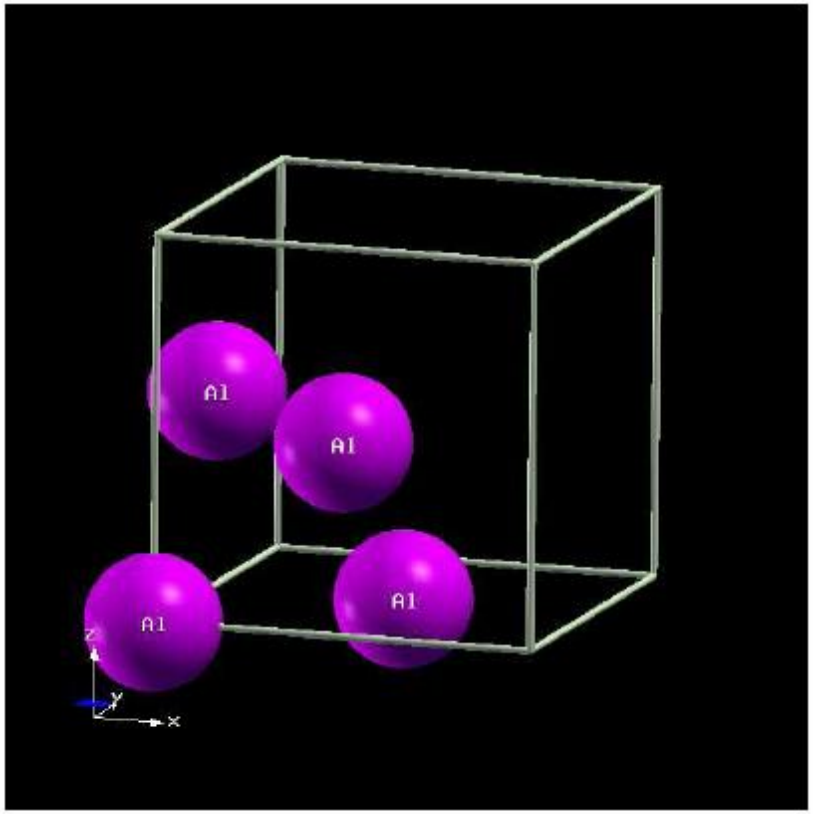
bulk Al: cubic *fcc* and *sc* bravais lattice





Example 2: Al (FCC)

- a cubic lattice
- **ibrav= 1, nat= 4**



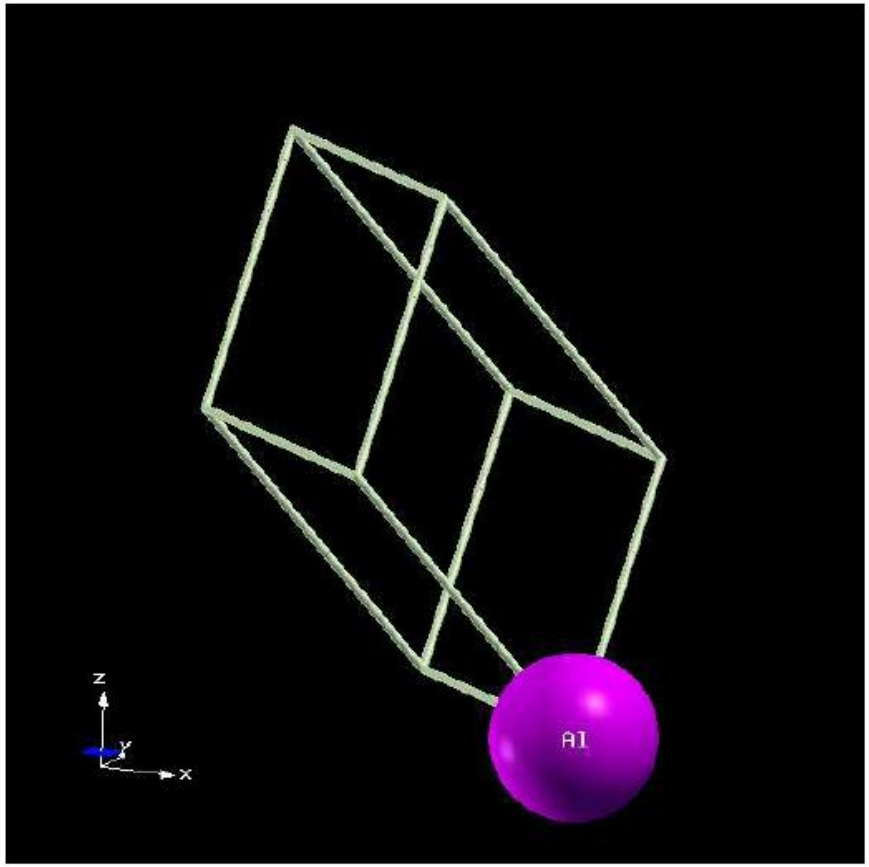
Cubic lattice

ATOMIC_POSITION:			
Al	0.0	0.0	0.0
Al	0.5	0.5	0.0
Al	0.5	0.0	0.5
Al	0.0	0.5	0.5

Example 2: Al (FCC)

FCC lattice

- an FCC lattice
- **ibrav= 2, nat= 1**



ATOMIC_POSITION:

Al 0.0 0.0 0.0

NOTE: Here, we show the **primitive** cell of the fcc structure in the left figure. This is because PWscf uses primitive cell for computation.

The primitive cell takes the most advantage of the symmetry of the crystal as it significantly reduces the number of unique k points

Example 2: Al (FCC)

- Let's revisit the coordinates in cubic cell.

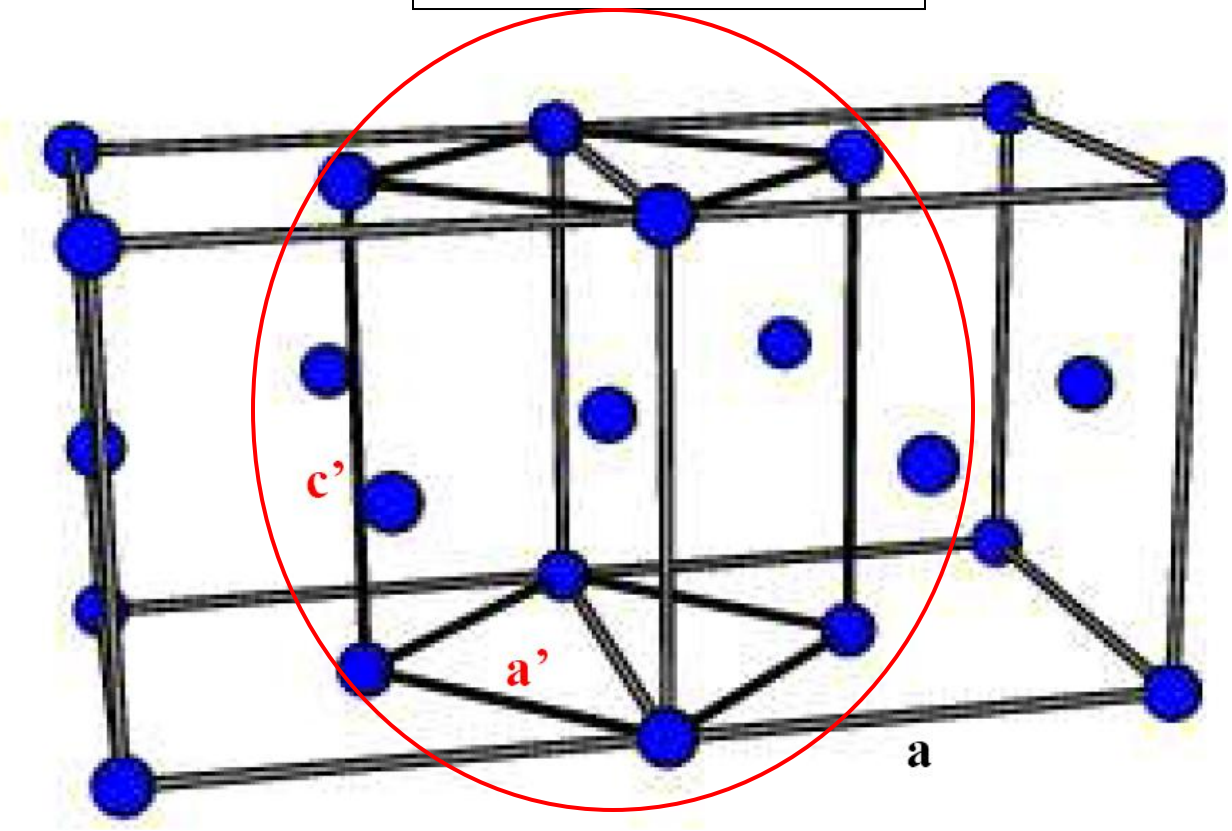
ATOMIC_POSITION:			
Al	0.0	0.0	0.0
Al	0.5	0.5	0.0
Al	0.5	0.0	0.5
Al	0.0	0.5	0.5

Recall, the default option for the ATOMIC_POSITION is *alat*. So what is *alat* ?

- If we assume the lattice constant is celldm(1) = a, then everything is in the unit of a. For example, (0.5, 0.5, 0) means the coordinate in the x direction is 0.5a, in the y direction is 0.5a, too. However, if we use *crystal* coordinate, then the coordinate in the x direction is $0.5a_1$, in the y direction is $0.5a_2$.
- Therefore, it is obvious that for cubic lattice system, the *alat* coordinate is the same as *crystal* coordinate.
- But for non-cubic lattice, these two are not the same. Let use the next example to further illustrate this point. We construct the fcc structure in a tetragonal unit.

Example 2: Al (FCC)

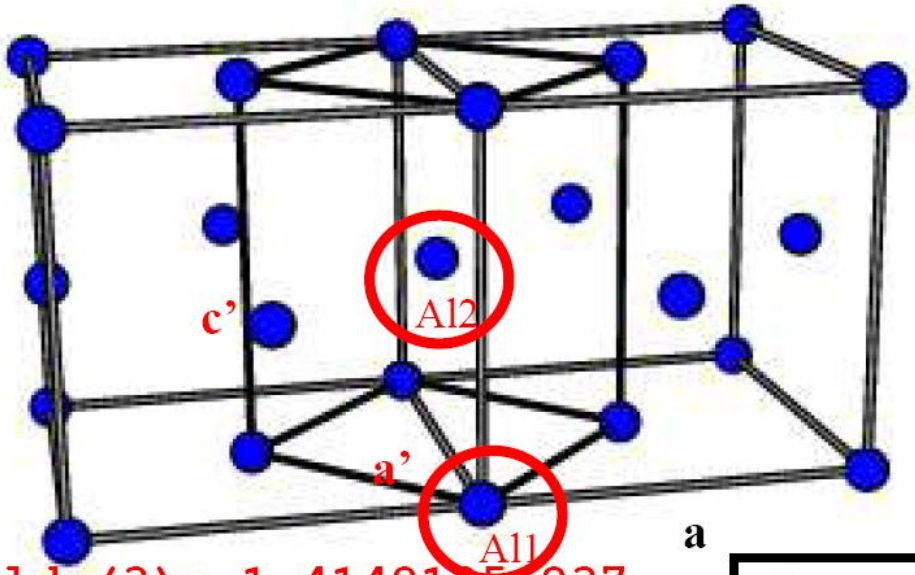
Tetragonal unit cell



It is easy to see from this figure that fcc structure can be also represented as a tetragonal unit cell due to the periodicity of the crystal structure.

Example 2: Al (FCC)

ATOMIC_POSTITION {alat}



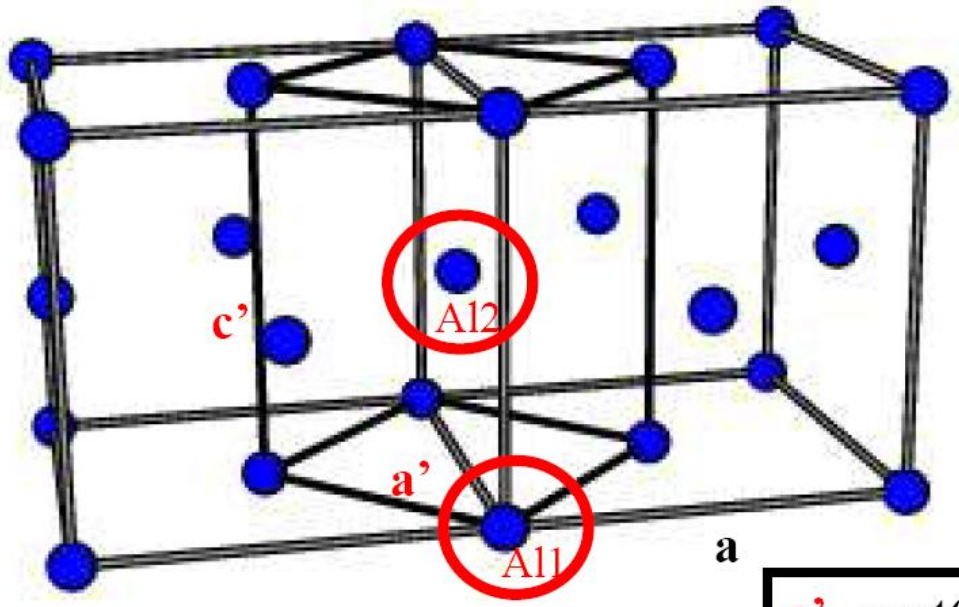

```
celldm(3)= 1.41421356237

Al  0.0  0.0  0.0
Al  0.5  0.5  0.707107
```

a'=sqrt(2)/2 * a
c'= a
c'/a'= sqrt(2)

ibrav= 6, celldm(1) =a, celldm(3) =1.41421356237
ATOMIC_POSITIONS {alat}
Al 0.00 0.00 0.00
Al 0.50 0.50 0.707107

Example 2: Al (FCC)

ATOMIC_POSTITION {crystal}



simple tetragonal (p)

==========

v1 = a(1,0,0),
v2 = a(0,1,0),
v3 = a(0,0,c/a)

```
Al 0.0 0.0 0.0
Al 0.5 0.5 0.5
```

a'=sqrt(2)/2 * a
c'= a
c'/a'= sqrt(2)

```
ibrav=  6, celldm(1) =a, celldm(3) =1.41421356237
ATOMIC_POSITIONS {crystal}
Al 0.00 0.00 0.00
Al 0.50 0.50 0.5
```

Energy Cutoff

- Remember that we are dealing with infinite systems using periodic boundary conditions. This means that we can use Bloch theorem to help us solve the Schrödinger equation. The Bloch theorem says:

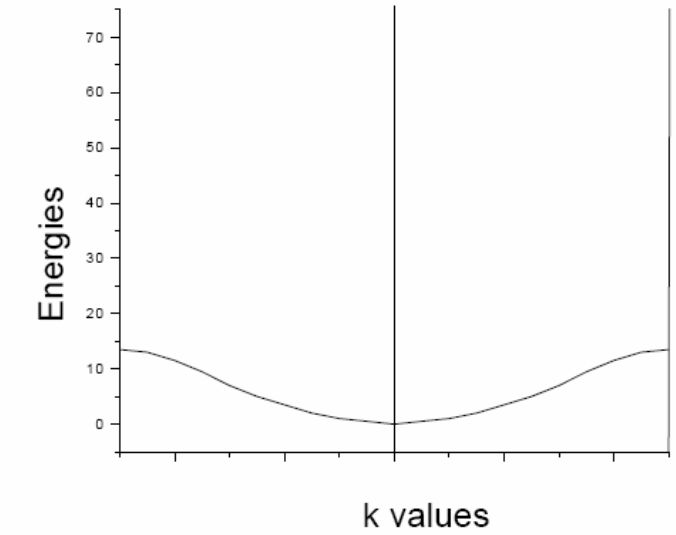
$$\psi_{n\mathbf{k}}(\mathbf{r}) = \exp(i\,\mathbf{k}\cdot\mathbf{r})\,u_{n\mathbf{k}}(\mathbf{r}) \quad \text{with} \quad u_{n\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{G}} \exp(i\,\mathbf{G}\cdot\mathbf{r})$$

In these equations. $\psi_{n\mathbf{k}}(\mathbf{r})$ s the wave function, $u_{n\mathbf{k}}(\mathbf{r})$ is a function that is periodic in the same way as the lattice. In this case, our *basis* functions (i.e. what we expand in) are plane waves.

- In actual calculations, we have to limit the plane wave expansion at some point . Or in other words, the wave function is only spanned on a finite basis of plane waves. This is called the *planewave cutoff*.
- Problem 1 is designed to test cutoff convergence issues. To test the convergence, you may choose the energy from largest cutoff value as the reference value, and calculate the difference between the calculate energy for smaller cutoff and this reference value.

k – point sampling

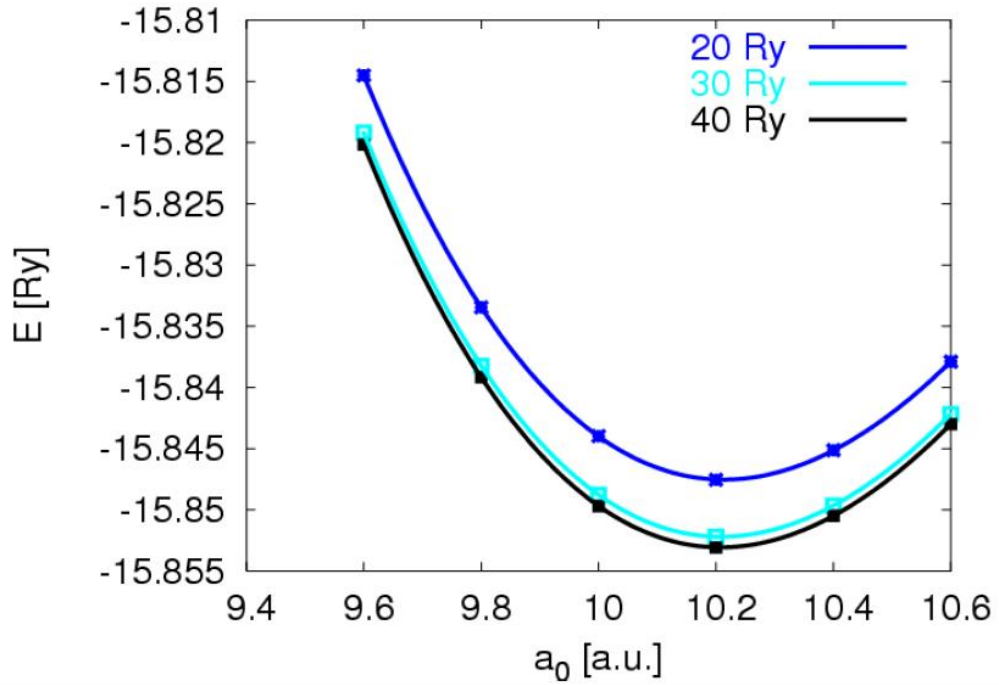
- Because of the Bloch theorem, we need to solve a Schrödinger-like Kohn-Sham equations (i.e. iterate self consistently the diagonalization of a MxM matrix, where M is the number of basis functions) everywhere in the Brillouin Zone.
- In practice, we do it for a finite number of k values , and get a value for E at each k. This is seen in the schematic below.



- To obtain a value for E, the energy of the crystal, we need to integrate over the first Brillouin Zone, where the bands are occupied. Thus summing over a finite number of k points is an approximation to performing an integral. You will need to make sure you have enough k-points to have a converged value for the energy.

First-principles calculations

- Overall, for all first-principles calculations, you must pay attention to two convergence issues.
- The first is *energy cutoffs*, which is the cutoff for the wave function expansion.
- The second is *number of k-points,* which measures how well your discrete grid has approximated the continuous integral.



Bulk Modulus

- Expand the energy in a Taylor series as a function of volume,

$$E(V) = E(V_0) + (V - V_0)\frac{dE}{dV} + \frac{(V - V_0)^2}{2}\frac{d^2E}{dV^2}$$

$P = -\frac{dE}{dV} = 0$ at equilibrium, so we have

$$E(V) = E(V_0) + \frac{(V - V_0)^2}{2}\frac{d^2E}{dV^2}$$

Now use $P = -\frac{dE}{dV}$, we have

$$E(V) = E(V_0) - \frac{(V - V_0)^2}{2}\frac{dP}{dV}$$

Now use $B = -V_0\frac{dP}{dV}$ where V_0 is the equilibrium volume. Finally, we have the expression for the bulk modulus:

$$E(V) = E(V_0) + \frac{(V - V_0)^2}{2V_0}B$$

We can fit the data to this equation to find B.